

Database Corruption



*Presented by Justin Feuerstein
October 24, 2000*

AMS

- Page and Object Allocation Concepts
- Database Consistency Checker (DBCC)
- Page allocation DBCC Commands
- Table Consistency DBCC Commands
- Other DBCC Commands
- Fixing the Corruption

- Allocation Unit - Units that new space is divided into when you initialize a database device
 - Consists of 256 2k Data Pages
- Allocation Page - First page of each allocation unit - tracks the use of all pages in the allocation unit -
 - All allocation pages have obj_id = 99
- Extent - Block of space allocated to new tables or indexes or when a table or index requires space
 - consists of 8 2k pages (32 extents in as allocation unit)
 - When a table or index is dropped, the extents it occupies are deallocated

Allocation Unit (256 2k Pages)

Allocation Page	0	1	2	3	4	5	6	7
	8	9	10	11	12	13	14	15
	16	17	18	19	20	21	22	23
Extent 24	24	25	26	27	28	29	30	31

	240	241	242	243	244	245	246	247
	248	249	250	251	252	253	254	255

32 Extents

- Each table and index has an OAM
- The OAM is stored on the first page of the first segment
- OAM Pages maintain information on rows and pages used by tables and indexes
- OAM pages point to the allocation page for each allocation unit where the object uses space
 - e.g., if `dsk_obj` is stored on extents 32 and 272, the OAM page for `dsk_obj` points to pages 0 and 256.
- The OAM is checked when a new page is needed for a table or index

- Each page contains a header that includes the number of the page that precedes it and the number of the page that follows it.
- When a new page is updated, the header information on the surrounding pages change to point to that page

A set of utility commands that are used to check the logical and physical consistency of a database.

- Page allocation DBCC commands check that:
 - Allocation pages contain valid information
 - All pages correctly allocated
 - No page is allocated that is not used
 - No page is used that is not allocated
 - Examines OAM page information
 - Reports on the number of pages and extents allocated to a table

- dbcc checkalloc - Checks to see if the page allocation for a database is consistent

dbcc checkalloc (database_name [,fix | nofix])

- dbcc indexalloc - Applies to a single index structure

dbcc indexalloc (table_name, index_id [{full | optimized | fast | null} [, {fix | nofix}])

- dbcc tablealloc - Applies to a single table structure

dbcc tablealloc (table_name [{full | optimized | fast | null} [, {fix | nofix}])

- dbcc textalloc - Checks the integrity of a table's text page chains

dbcc textalloc ("all" | table_name [{full | optimized | fast | null} [, {fix | nofix}])

- **full**: Checks all pages for all types of allocation errors.
(method used by dbcc checkalloc)
- **optimized**: Checks only allocation pages referenced in the OAM pages.
- **fast**: Checks whether all data or index pages that are referenced as part of a page linkage have been allocated
- **fix**: Fixes allocation errors detected by the consistency checker.
- **nofix**: Does not fix any allocation errors.

- Reports the number of pages and extents that have been allocated to a table and any non-clustered indexes

```
TABLE: proc_object          OBJID = 1471344306
INDID=1      FIRST=64289      ROOT=64281      SORT=1
      Data level: 1.  410 Data pages allocated and 52 Extents
allocated.
      Indid    : 1.   4 Index pages allocated and 1 Extents
allocated.
INDID=2      FIRST=64274      ROOT=64273      SORT=1
      Indid    : 2.   38 Index pages allocated and 5 Extents
allocated.
INDID=3      FIRST=64298      ROOT=64297      SORT=1
      Indid    : 3.   39 Index pages allocated and 5 Extents
allocated.
INDID=4      FIRST=64306      ROOT=64305      SORT=1
      Indid    : 4.   98 Index pages allocated and 13 Extents
allocated.
INDID=5      FIRST=64314      ROOT=64313      SORT=1
      Indid    : 5.   87 Index pages allocated and 11 Extents
allocated.
TOTAL # of extents = 87
```

- Checks all allocation pages in a database to ensure that they contain valid information

```
Alloc page 0 (# of extent=32 used pages=128 ref pages=118)
Alloc page 256 (# of extent=32 used pages=126 ref pages=123)
Alloc page 512 (# of extent=32 used pages=196 ref pages=188)
Alloc page 768 (# of extent=32 used pages=206 ref pages=206)
Alloc page 1024 (# of extent=32 used pages=183 ref pages=183)
Alloc page 1280 (# of extent=32 used pages=211 ref pages=179)
Alloc page 1536 (# of extent=32 used pages=221 ref pages=189)
Alloc page 1792 (# of extent=32 used pages=221 ref pages=221)
Alloc page 2048 (# of extent=32 used pages=249 ref pages=241)
Alloc page 2304 (# of extent=32 used pages=256 ref pages=256)
Alloc page 2560 (# of extent=32 used pages=250 ref pages=178)
Alloc page 2816 (# of extent=32 used pages=256 ref pages=0)
Alloc page 3072 (# of extent=32 used pages=256 ref pages=0)
Alloc page 3328 (# of extent=32 used pages=256 ref pages=0)
Alloc page 3584 (# of extent=32 used pages=256 ref pages=0)
Alloc page 3840 (# of extent=32 used pages=256 ref pages=0)
Alloc page 4096 (# of extent=32 used pages=256 ref pages=0)
Alloc page 4352 (# of extent=32 used pages=256 ref pages=0)
Alloc page 4608 (# of extent=32 used pages=256 ref pages=0)
Alloc page 4864 (# of extent=32 used pages=256 ref pages=0)
Alloc page 5120 (# of extent=32 used pages=256 ref pages=0)
Alloc page 5376 (# of extent=32 used pages=256 ref pages=0)
Alloc page 5632 (# of extent=32 used pages=256 ref pages=0)
```

- Examines Object Allocation Map (OAM) page information

The number of pages used and unused for object 973246522 index 0 on allocation page 1302528 do not match the counts in the OAM entry.

The counts in the OAM are incorrect. This implies that there are entries missing. Run tablealloc utility with the FIX option on the table with the inaccurate OAM counts.

- No page is allocated that is not part of a page linkage - No page is allocated that is not used

Table Corrupt: Page is allocated but not linked; check the following pages and ids: allocation pg#=36608
extent id=36688 logical pg#=36695 object id on
extent=983674552 (object name = dsk_lock) indid on
extent=0

Table Corrupt: Page is allocated but not linked; check the following pages and ids: allocation pg#=598528
extent id=598544 logical pg#=598545 object id on
extent=8 (object name = syslogs) indid on extent=0

- No page is part of a page linkage that has not been marked as allocated - No page is used that is not allocated)

Table Corrupt: Page is linked but not allocated.
Run DBCC TABLEALLOC to correct the problem.
(alloc page#=645120, extent id=645120, logical
page#=645121, object id in extent=0, index id in
extent=0, object name=0)

Table Corrupt: Page is linked but not allocated.
Run DBCC TABLEALLOC to correct the problem.
(alloc page#=612096, extent id=612336, logical
page#=612338, object id in extent=8, index id in
extent=0, object name=syslogs)

- Checks all allocation pages in a database to ensure that they contain valid information OR All pages have been correctly allocated

```
Table Corrupt: Object id wrong; tables: alloc
page 858880 extent id=858936 1 page#=858937
objid in ext=268580045 (name = clin_resp_xref)
objid in page=909246294 (name = legal_text_data)
objid in sysindexes=268580045 (name =
clin_resp_xref)
```

```
Table Corrupt: Object id wrong; tables: alloc
page 82944 extent id=83184 1 page#=83191 objid
in ext=1319675749 (name = dsk_obj_hist) objid in
page=541244983 (name = ldi_text) objid in
sysindexes=1319675749 (name = dsk_obj_hist)
```


Table Corrupt: Extent id 633016 on allocation pg# 632832 has objid 8 and used bit on, but reference bit off.

Table Corrupt: Extent id 256056 on allocation pg# 256000 has objid 0 and used bit on, but reference bit off.

Table Corrupt: Attempted to get page 86845, object 1319675749; got page 0, object 0.

- Data pages and indexes are properly linked - Is the previous page, current page, and next page consistent throughout the page linkage?
- Indexes are sorted properly in the correct order
- Consistency of all pointers - Are index pointers to the page and row still valid?
- Data rows on each page had entries in the OAM page and they match their respective locations on the page.
 - Updates the information on the OAM pages that potentially can become inaccurate.

- Table Consistency Commands checks page linkages and data pointers at both the row and page level.
- dbcc checkdb - Checks table consistency for an entire database
`dbcc checkdb (database_name[,skip_ncindex])`
- dbcc checktable - Checks table consistency for a single table
`dbcc checktable (table_name | table_id> [,skip_ncindex])`

skip_ncindex is an optional parameter that enables you to skip checking non-clustered indexes

Checking ldi_text

The total number of data pages in this table is 444.

The total number of TEXT/IMAGE pages in this table is 210507.

Table has 14981 data rows.

Checking sys_pbuo

The total number of data pages in this table is 6.

Table has 179 data rows.

Checking corr_log_detail

The total number of data pages in this table is 1.

Table has 9 data rows.

Checking pbcattbl

The total number of data pages in this table is 2.

Table has 69 data rows.

- Indexes are sorted properly in the correct order

Keys of index id 1 for table 'usr_security' in data page 339584 should be in ascending order. Drop and re-create the index in database 'SPS_M67001_DB'.

Key mismatch between index page 305312 (row 48) and data page 308727 in database 'SPS_M67001_DB'. Drop and re-create index id 2 of table 'place_ref'.

Server Message: Number 605, Severity 16 Line 1:

An attempt was made to fetch logical page '858937' in database 'M4PD2PP1' from cache 'default data cache'. Page belongs to object with id 909246294', not to object 'clin_resp_xref'.

Server Message: Number 625, Severity 16 Line 1:

Could not retrieve row from logical page 36689 via RID because the entry in the offset table for that RID is less than or equal to 0.

Table Corrupt: key(s) in the left child is not less than the parent key; check left child page 866006 of index id 1 for table 'dsk_obj_audit' in database 'M4PD2PP1'.

Table Corrupt: The values in adjust table should be in ascending order starting from the end of the table (page#=15997 row#=9); check adjust table in this row

Table Corrupt: Row length is inconsistent between the computed row length and the recorded row length on page; check the following page and row: pageno=15997 row#=9 computed row length=345 row length on page=88

- Error message from within PD²



- Error message from Sybase error log

```
01:2000/05/30 09:55:39.52 server An attempt was made to
fetch logical page '757792' in database 'U4PD2PP1' from
cache 'default data cache'. Page belongs to object with
id '1015674666', not to object 'dsk_obj_hist'.
```


Key mismatch between index page 495335 (row 28) and data page 702098 in database 'U4PD2PP1'. Drop and re-create index id 2 of table 'dsk_configuration'.

Table Corrupt: Page linkage is not consistent; check the following pages: (current page#=757338; page# pointing to this page=757338; previous page# indicated in this page=757339)

Key mismatch between index page 36676 (row 0) and data page 36695 in database 'U4PD2PP1'. Drop and re-create index id 2 of table 'dsk_lock'.

Server Message: Number 625, Severity 16

Line 1:

Could not retrieve row from logical page 36695 via RID because the entry in the offset table for that RID is less than or equal to 0.

Table Corrupt: Page linkage is not consistent; check the following pages: (current page#=757747; page# pointing to this page=757747; previous page# indicated in this page=757748)

Server Message: Number 624, Severity 16

Line 1:

Attempt to retrieve row from page via RID failed because the requested RID has a higher number than the last RID on the page. Rid pageid = 0xb9018; row num = 0x4. Page pointer = 0xe590000, pageno = 757784, status = 0x105, objectid = 1239675464, indexid = 0, level = 0.

Server Message: Number 605, Severity 16

Line 1:

An attempt was made to fetch logical page '757792' in database 'U4PD2PP1' from cache 'default data cache'. Page belongs to object with id '1015674666', not to object 'dsk_obj_hist'.

The total number of data pages in this table is 1.

DBCC execution completed. If DBCC printed error messages, contact a user with System Administrator (SA) role.

- dbcc checkcatalog - Checks for consistency problems between system tables and within a system table in a database. It performs the following types of checks for all of the system tables within a database:
 - Verifies a table or view in the sysobjects table has at least one row in the syscolumns table
 - Verifies that a type in the syscolumns table has a row in the systypes
 - Checks that the last checkpoint in the syslogs table is valid
 - Checks the segment definitions in the syssegments table

Syntax: **dbcc checkcatalog (database_name)**

Checking master

The following segments have been defined for database 1
(database name master).

virtual start addr	size	segments
-----	-----	-----

4	1536	0 1 2
3588	1024	0 1 2

DBCC execution completed. If DBCC printed error messages,
contact a user with System Administrator (SA) role.

- **dbcc{traceon | traceoff} (traceflag)**
 - **302** – Prints information on index selection
 - **310** – Prints information on join selection by selecting the first plan chosen and all cheaper plans.
 - **317** – Prints complete information on join selection by selecting the first plan and all more expensive plans.
 - **1204** – Prints deadlock information. Can also use **sp_configure “print deadlock information”, 1**
 - **3604** – Directs output from dbcc commands to window where the command is executed
 - **3605** – Directs output from dbcc commands to the server error log.

- Dbcc page - Enables you to view a page header and, optionally, the data on a page

```
dbcc page( dbid|dbname, pagenum [, display_option={0|1|2|3} [,  
cache={0|1} [, logical={1|0} [, cachename | -1 ]]] )
```

- Printopt

- 0 = displays only page header without the page contents (default)
- 1 = displays the page header and a hex dump of the page contents individually by row
- 2 = displays the page header and the page contents in a single block of data
- 3 = Used when displaying the contents of a control page for a partitioned table

- cache

- 0 = fetch the page from disk
- 1 = fetch the page from memory (default)

- logical

- 0 = virtual page
- 1 = logical page (default)

PAGE HEADER:

Page header for page 0x20652000

pageno=152836 nextpg=152837 prevpg=152835

objid=653245382 timestamp=0001 00788b4f

nexttrno=32 level=255 **indid=255** freeoff=1832

minlen=0

page status bits: 0x20 (0x0020 (PG_TEXT))

Indid

0 = data page, 1 = clustered index, 2-250 =
nonclustered index, 255 = text/image page

- Drops a damaged database
 - Use if the drop database command does not work

Syntax: `dbcc dbrepair (database_name, dropdb)`

- Displays the syntax for a dbcc command

Syntax: **dbcc help (dbcc command)**

Example:

```
dbcc help (tablealloc)
```

```
tablealloc(tablename|tabid, [full |  
optimized | fast],[fix | nofix])
```


- DBCC pglinkage
 - Displays the page chain and checks the consistency of page pointers for a table or index page chain.
- DBCC memusage
 - The current allocation of memory within SQL server
 - Up to the 20 largest tables and/or indexes currently in data cache
 - Up to the 20 largest stored procedures currently in procedure cache.

- Always check the Sybase Error log for additional errors to try to determine what may have caused the error
- Check System error logs such as the event viewer to determine if system problems occurred
- Some errors may be caused by the dbcc commands being run while users are in the database and they may not be real errors
- Search DBCC output for following terms:
 - ‘Server Message’
 - ‘Corrupt’
 - ‘Drop and re-create’
 - ‘Msg’

- Page allocation problems - run dbcc tablealloc with fix
- Restore database from backup
- Indexes - drop and recreate
- Rebuild table by using 'select into'
- Rebuild table by using bcp
- Rebuild table from its corresponding audit table

Always search the Sybase knowledge base at www.sybase.com/support